//分别利用普通写法和多态技术实现计算器

//多态优点

//利于前期后期拓展及维护

//可读性强

//代码组织结构清晰

#include<iostream>

#include<string>

using namespace std;

//普通写法

class cal {

public:

int getresult(string oper) {

if (oper == "+") {

return mnum1 + mnum2;

}

else if (oper == "-") {

return mnum1 - mnum2;

}

else if (oper == "\*") {

return mnum1 \* mnum2;

}

//如果想要拓展新的功能，需要修改源码

//在真实开发中，提倡开闭原则

//开闭原则：对拓展进行开放，对修改进行关闭

}

int mnum1;

int mnum2;

};

void test01() {

cal c;

c.mnum1 = 10;

c.mnum2 = 10;

cout << c.mnum1 << "+" << c.mnum2 << "=" << c.getresult("+") << endl;

cout << c.mnum1 << "-" << c.mnum2 << "=" << c.getresult("-") << endl;

cout << c.mnum1 << "\*" << c.mnum2 << "=" << c.getresult("\*") << endl;

}

//利用多态实现计算器

//实现计算器的抽象类

class abstractcal {

public:

virtual int getresult() {

return 0;

}

int mnum1;

int mnum2;

};

//加法加速器

class addcal :public abstractcal {

public:

virtual int getresult() {

return mnum1+mnum2;

}

};

//减法计算器

class subcal :public abstractcal {

public:

virtual int getresult() {

return mnum1 - mnum2;

}

};

//乘法计算器

class mulcal :public abstractcal {

public:

virtual int getresult() {

return mnum1 \* mnum2;

}

};

void test02() {

//多态使用条件

//父类指针指向子类对象

//加法运算

abstractcal\* asd = new addcal;

asd->mnum1 = 10;

asd->mnum2 = 900;

cout << asd->mnum1 << "+" << asd->mnum2 << "=" << asd->getresult() << endl;

//记得销毁

delete asd;

//减法运算

asd = new subcal;

asd->mnum1 = 120;

asd->mnum2 = 243;

cout << asd->mnum1 << "-" << asd->mnum2 << "=" << asd->getresult() << endl;

delete asd;

//乘法运算

asd = new mulcal;

asd->mnum1 = 20;

asd->mnum2 = 3;

cout << asd->mnum1 << "\*" << asd->mnum2 << "=" << asd->getresult() << endl;

delete asd;

}

int main(){

//test01();

test02();

system("pause");

return 0;

}